E-908

A METHOD AND SYSTEM FOR MULTI-CARRIER PACKAGE TRACKING

Cross Reference to Related Application(s):

Reference is made to U.S. application, Serial No. ______________________________ (Attorney Docket No. E-907) , filed on even date herewith, entitled, A METHOD AND SYSTEM FOR ESTABLISHING PARCEL SHIPPPING VIA THE INTERNET; assigned to the assignee of this application,

Reference is made to U.S. application, Serial No. ______________________________ (Docket No. E-909), filed on even date herewith, entitled, A METHOD AND SYSTEM FOR RESOLUTION OF CARRIER SPECIFIC DATA UTILIZING A GENERIC DATA MODEL, assigned to the assignee of this application.. The subject matter of each of these applications is hereby incorporated ~~by reference.~~

Technical Field:

The present invention relates to package tracking, particularly to the tracking of a package enroute from the sender to the recipient, wherein the package may be delivered by one or a plurality of package carriers.

Background of the Invention:

In a shipping system such as set forth in co-pending application, U.S. Serial No. 09/411,092 filed Oct. 4, 1999 entitled, A Method and System for Establishing Parcel Shipping Via The Internet ~~(E-907)~~, one of the components thereof is the ability to be able to track the location of a package while it is transit from the sender (user of the shipping system) to the recipient. It is desirable to be able to do this tracking regardless of the particular carrier which is

2

actually delivering the package, as well as to be able to provide additional services and features concerning package location and delivery.

Summary of the Invention:

In the package tracking system and method according to the present invention, the user (sender) of a package is able to determine the package's location while enroute to the recipient. In order to achieve this result, the present invention uses network based services and in particular, the Internet to provide the means for transferring tracking information from the carrier responsible for delivery of the package to the party requesting information, typically the sender. In particular, the architecture of the present invention comprises tracking objects which are transferred to the carrier's tracking website in a manner so as to properly request information with regard to an identified package. The shipping system server is responsible for creating the tracking objects, as well as for accessing the selected carrier's tracking website. In particular, the tracking objects are created and their creation and transfer controlled by a tracking coordinator forming part of the shipping system's server. The tracking coordinator performs these tasks through use of registry settings associated with the operating system under which the shipping system server is operating.

The tracking coordinator of the shipping system's server controls these objects by timing the initiation of delivery of the objects to the carrier, as well as the frequency of these deliveries so as to comply with carrier requirements concerning requests for tracking information. The overall result is that tracking of a package is obtainable in a straight-forward fashion and in a manner which does not overload the carrier tracking website.

Brief Description of the Drawings:

For a fuller understanding of the nature and objects of the present invention, reference is made to the following detailed description taken in conjunction with the following drawings in which:

Figure 1 is a diagrammatic representation of the shipping system and method which may use the tracking system and method of the present invention.

Figure 2 is a flow chart of a typical sequence of shipping steps associated with a shipping system and method in which the tracking system of the present invention forms a part thereof.

Figure 3 is an overall block diagram showing the typical architectural implementation of the shipping system and method of which the tracking system forms a part thereof.

Figure 4 is a more detailed block diagram of the architecture for the shipping system of which the tracking system of the present invention forms a part thereof.

Figure 5 is a diagrammatic representation of the tracking system and method, illustrating the multi-threaded implementation of the program execution.

Figure 6 is a block diagram illustrating the use of multiple Internet Protocol (IP) addresses to increase the number of tracking objects while observing tracking object constraints set by a carrier.

Figure 7 is a flow chart of the tracking system and method shown in Figure 5.

<u>Detailed Description:</u>

Art 02 The tracking system and method of the present invention forms part of an overall shipping system and method as described in copending, U.S. application, Serial No. ____________________ filed ____________________ entitled, A Method and System for Establshing Parcel Shipping Via the Internet (E-907); and U.S. application, Serial No. ____________________ filed

4

__________________ entitled, A Method and system for Resolution of Carrier Specific Data Utilizing a Generic Data Model (E-909). As best seen in Figure 1, this overall shipping system **20** comprises a shipping system server **22** and one or more users (senders) **26** which interact with the server by means of the Internet **24**, typically through connection through an Internet service provider **28**. Although a plurality of users at a single location are shown which are interconnected by a local area network **30**, the system also allows other users (e.g. users **26** and **26′**) to access the shipping system via the Internet.

Figure 2 illustrates the sequence of steps and sub-steps executed in order for a package to be shipped using the shipping system and method. Thus between the begin shipping step **31** and the shipping complete determination step **49**, there can be a number of steps including the determination of the rate associated with sending the package, step **32**, the determination of the address of the recipient, step **34**; the determination of the weight of the package, step **36**; options associated with the package, step **38**, such as a possible insurance sub-step **39**, the recording of payment sequence step **40**; the printing of labels for the package, step **42**; the recording of data associated with the package, record package data sequence step **44**; whether a request for pick-up is made, request pick-up step **46**; and the generation of E-mail requests including requests regarding the scheduling of tracking and the like, step **48** and sub-steps **51** for schedule tracking and the sending of e-mail information to the system service database, sub-step **53**.

In order to implement these sequence of steps, the shipping system and method uses a multi-tier architecture such as shown in Figures 3 and 4. The tracking component **88** and InstaTrac component **89** are best illustrated in Figure 5 with respect to their operations. Figure 5 is a diagrammatic representation of the tracking system (comprising components **88**, **89**, **98** and **138**). In operation, a user **26** desiring tracking information makes such a request over the Internet to the shipping system server **22** (Figure 1). This shipping server operates so as to generate a tracking object which is a component object model (COM) created by the InstaTrac component **89**. The task of this tracking object is to initiate an Internet request to the selected carrier's tracking website **130**, where each carrier has its own tracking component associated with its web based services. The shipping system

server tracking component **88** controls sending tracking objects to the designated carrier tracking website. In order to do this, the tracking number created at the time that the package was shipped (see Figure 2 for sub-step **41**) is used to identify the package to the corresponding carrier tracking website **130**. The creation of these tracking objects must be paced since various carriers restrict the number of tracking objects that can be sent to the carrier tracking website. For example, a particular carrier tracking website may restrict the number of requests from a particular Internet Protocol (IP) address to one request every ten seconds. Thus, if the carrier sees tracking objects more frequently than one every ten seconds, it may interpret the tracking objects as an attack on the tracking website, resulting in lockout of the IP address.

In addition, other carriers may permit only a fixed number of tracking objects within a specified period of time, without regard to the frequency of such requests. It is thus the obligation of the shipping server tracking coordinator **121** to ensure that tracking objects are controlled with respect to the frequency of their generation in view of the requirements of each of the carrier tracking websites. As seen in Figure 5, this is accomplished by the tracking coordinator **121**. The tracking coordinator generates tracking objects **123** for delivery over the Internet **24** to the associated carrier tracking website **130**. As seen in Figure 5, tracking objects are designated by the tracking coordinator **121** for each carrier in which tracking information is desired. The tracking coordinator obtains information for these tracking objects from corresponding carrier input tracking request queue **119**. For each tracking object within a tracking request queue, the tracking coordinator obtains that request through an associated programming thread since the tracking coordinator operates in a multi-threaded manner.

Depending upon the pacing constraints set by the carrier, multiple threads for generating multiple tracking objects are generated. Thus for instance as shown in Figure 5 for tracking objects to be sent to the Airborne tracking website **130**, multiple Airborne tracking objects **123′** can be generated. In this example, Airborne may have a constraint that no more than five tracking objects can be generated within a pre-determined length of time and these tracking objects all would be then generated (assuming that at least that number of tracking requests are in the Airborne tracking

request queue **119′**) by the tracking coordinator on individual software threads **125**. Similarly, the tracking coordinator generates tracking objects via individual threads for each of the carriers supported by the shipping system server, which in the example shown in Figure 5 comprises four carriers; namely, Airborne, UPS, FedEx and DHL.

As also seen in Figure 5, a user of the shipping system can request an instant track regarding a package for which the tracking number is known by the user. This is shown by InstaTrac component **89** which is executed by the shipping server (see Figure 1) and which is conveyed to the appropriate carrier input tracking request queue **119** as shown by line **127**. Such a tracking request from a user will generally take priority over other tracking requests that are periodically received from the tracking component **88** and which are generated in an automatic fashion by the shipping server so as to maintain tracking information for all of the packages handled by the shipping server. This prioritization is accomplished by placing the InstaTrac request at the top of the associated queue **119**.

The tracking objects as discussed above are conveyed to the corresponding carrier tracking website **130** via the Internet **24** as shown by arrows **129**. The carrier tracking website **130** then processes each tracking request from the tracking object and generates a tracking response which is conveyed as an HTML page **131**. The HTML page contains the tracking status and is transferred via the Internet **24** to the tracking component **88**. The tracking component then parses the HTML and extracts the tracking request results. The results are then sent to the tracking results queue **133**. The tracking coordinator via path **137** senses if tracking results data is in the tracking result queue. If data is sensed, the tracking coordinator via path **139** causes the business component **138** to receive the data from the tracking results queue. This component is used to update the data services module **98** in the groupware services tier **64** (see Figure 4) so as to convey the updated tracking information to the proprietary database **74**. Tracking coordinator **121** controls the business component **138** via control line **139**.

Thus an important aspect of the tracking coordinator is its multi-threaded executable architecture which allows for simultaneous generation of tracking objects for various carriers, all within the pacing constraints associated with the individual

7

carriers. Thus the tracking coordinator operates as a multi-threaded service. It accepts commands from the scheduling component **94** which is used to initiate the autonomous or automatic generation of tracking requests for maintaining updated tracking information concerning outstanding packages.

Details of the Tracking Coordinator:

The tracking coordinator **121** gathers its inputs from the carrier input tracking request queues **119**. Each carrier is seen to have a corresponding queue. The names of each input queue is stored in the operating system Registry, along with other carrier information. For each carrier found within the Registry a thread **127** is generated. This thread monitors the corresponding carrier input queue for incoming messages.

The tracking request messages stored in the input queues **119** are deposited by the scheduling module **94** (see Figure 4) or by the InstaTrac module **89**.

The InstaTrac module creates a tracking request from an ASP request and then to the associated carrier input queue **119** as shown by arrow **127**. Such an InstaTrac tracking request contains a record set and the tracking coordinator extracts the tracking request from the queue as soon as it is placed there, thereby prioritizing the InstaTrac tracking request. The tracking coordinator then extracts the record set from the tracking request and invokes an outgoing tracking object thread **125** that activates the corresponding carrier's tracking component at the carrier's tracking website **130**. The tracking coordinator sends several parameters to the carrier's tracking component, one of which is the record set that contains information pertinent to the particular tracking request, regardless of the origin of the tracking request.

The record set in general may contain one or more records. If the record set contains a single record, the tracking coordinator invokes a single tracking object and provides the record set as a parameter of that object. This mode of operation is called the "same mode". If however the tracking request contains a record set with many records, referred to a "mixed mode", the tracking coordinator separates the record set and dispenses the individual records as individual tracking objects. The

individual tracking objects are then distributed in a timely manner in order to meet the specified pacing requirements of the associated carrier.

The tracking coordinator only creates tracking object threads if it is allowed to place another tracking object within a specified time period. This time frame information which is keyed to the individual carrier is again stored in the operating system Registry. Thus for instance, if the carrier's tracking website only allows for a particular IP address to send two requests in a one minute time interval, this information is stored in the Registry. Then when the tracking coordinator generates tracking objects, it obeys this constraint by invoking threads only within that time constraint. In this manner, the carrier's tracking website does not deny access by these tracking objects since they are generated at a pace which is within the constraint set by that carrier.

As also seen in Figure 5, the tracking coordinator via control line **139** launches the business component software module. This component, as explained earlier, invokes the data services module **98** which is responsible for updating the tracking information stored within the client (user) database **76**.

Since the tracking coordinator operates as a Microsoft Windows NT service, it is started from a service control program. The service's icon within the control panel is an SCP that can manipulate the states of a service.

Once the service installs into the Registry (HKEY_LOCAL_MACHINE\YSTEM|CurrentControlSet\Services) it is manipulated via a SCP. Preferably this service has its startup mode set to automatic. This allow the tracking coordinator to start upon system startup, thus providing continuous service. Table 1 provides the commands for Registry installation and removal of the service.

Table 1

tracking coordinator - install	Installs the service into the Registry.
tracking coordinator - remove	Removes the service from the Registry.

T0090

This service supports the Start, Stop, Pause and Continue commands.

Start

Begins the carrier threads.

Stop

stops all carrier threads and waits for any tracking component threads to exit before stopping the service. This command may take up to 30 seconds to complete if there are several tracking component threads still running.

Pause

This command suspends all carrier threads currently running.

Continue

This command resumes any carrier threads that may have been suspended due to a Pause command.

System Requirements

The tracking coordinator needs to access the carrier input queues **119**. These queues are made available on the same domain. Further, in order to access any of the queues their names are made available through the Registry.

Compiler & Linker Requirements

This component is created with Visual Studio C++6.0.

In order to debug this component it is sometimes necessary to not run it as a service. The command of tracking coordinator -debug can be used to run the tracking coordinator from a Command Prompt window. In this mode the component runs without any interaction from the SCM.

Error Logging

All error and informational messages are written to the NT system application event log.

Registration Details

The following registry entries must be provided. Failure to furnish any one of the key or string values results in the service not being able to start. The failure is logged to the Systems Event View, Application Log.

[HKEY_LOCAL_MACHINE\SOFTWARE\PitneyBowes\PSS]
"MSMQServer"="Glenserver\\"
"ResultsQ"="trackingresultsq"

Each carrier that is monitored must have the following entry with the corresponding string values.

[HKEY_LOCAL_MACHINE\SOFTWARE\PitneyBowes\PSS\Carriers]
[HKEY_LOCAL_MACHINE\SOFTWARE\PitneyBowes\PSS\Carriers\49]
"Trac_URL"="208.243.33.179/tracktest/default.asp"
"RequestQ"="upsrequestq"
"ObjectsPerTimeFrame"="2"
"RecordsSetMode"="Same"
"RecsPerRecSet"="50"
"Symbol"="UPS"
"TimeFrameUnits"="min"
"Token"="49"

Registry String Value Definitions

Registry string values as set forth in Table 2.

Table 2

T0110

MSMQServer: This is the name of the server where the tracking queues exist.
ResultsQ: Name of the queue that a tracking component dumps its results.
Trac_URL: The URL that the tracking component hits.
RequestQ: Name of the carrier input queue **119** that the tracking coordinator extracts requests.

ObjectsPerTimeFrame: This is the number of tracking components that the tracking coordinator allows per time frame (i.e. minutes or hour).

RecordSetMode: Each message that is placed on a carrier queue contains a record set. A record set may contain one or many records. If the value of this string value is "Same", the record set contains only one record. If the value is "Mix", the record set can contain many records.

RecsPerRecSet: Maximum number of records a message's record set can contain.

Symbol: The carrier's symbol. This symbol is used to form the tracking component's OLEProgID (i.e. PSSUPS.ParseUPS).

TimeFrameUnits: The time frame that a specified number of Tracking Components can be invoked (i.e. minutes or hour).

Token: The carrier's token.

Figure 7 represents a flow chart of the steps taken by the shipping server to determine whether a package has been delivered by a carrier to the intended recipient. Step **190** shows that the shipping server initiates a tracking request to a specified carrier, depending upon date and time constraints. Thus a specified carrier such as United Parcel Service may only have guaranteed delivery at a specified time in the morning and a specified time in the afternoon. In order to make the request for tracking information efficient, it is therefore contemplated that tracking requests be made after these guaranteed times are exceeded. Thus for instance, if morning delivery has a guaranteed time of ten thirty A.M., a tracking request (via a tracking object) to UPS may be made at eleven A.M. to determine which packages have in fact been delivered within the ten thirty time constraint. Similarly, another request could be made after four thirty P.M. if afternoon guaranteed delivery is to be made by that time. Of course, different carriers can have different constraints, both with regard to guaranteed time of delivery as well as whether deliveries are guaranteed on non-business days such as Saturdays and Sundays.

Step **192** shows that the next step performed is retrieving tracking numbers from the client database **76** (see Figure 4) for packages that are being shipped by the specified carrier and which are not marked as delivered. Step **194** shows that the tracking coordinator **121** observes pacing constraints set by the specified carrier.

Thus, for instance, a specified carrier may demand that no more than a given number of tracking objects be made from a particular Internet protocol (IP address), such as the IP address of the shipping server within a given period of time. This might be limited to ten tracking objects per minute or no more than a hundred tracking objects in a thirty minute period, regardless of how many are presented in any time interval within that thirty minute period. In order to partially overcome the constraints of the specified carrier, the tracking coordinator can initiate tracking objects from a plurality of server sites such as shown in Figure 6, wherein each server site **196**, **197**, **198** has an associated unique IP address. In so doing, the carrier tracking website **130** will not interpret these additional tracking objects as an attack on the website since the tracking objects are emanating from unique IP addresses. Figure 6 therefore shows multiple server sites **196**, **197** and **198** so as to generate tracking objects from three unique IP addresses.

Referring again to Figure 7, the tracking objects are then sent by the shipping server from one or more IP addresses to the specified carrier tracking website. At Step **198**, the carrier tracking website has processed the tracking object and has created an HTML page that contains the tracking results for the package. These tracking results typically indicate if the package has been delivered to the recipient, and if so, the date and time of delivery. These results also typically show the dates, times and places the package has traversed in its path toward the recipient.

At step **199**, the carrier's tracking results HTML page is parsed by the tracking component. This is typically performed by "scraping" the HTML page, thereby extracting the relevant tracking information.

Decisional step **200** then determines whether the tracking results indicate that the package has been delivered to the recipient. If the package is still enroute, the tracking results are stored in database **76**, but the tracking number is not marked as "DELIVERED" (Step **202**). This tracking number therefore is available for subsequent rescheduling to the appropriate input tracking request queue **119**.

If the tracking results indicate that the package has been delivered, then that information is used to show that the package has in fact been delivered with that response then being used to invoke decisional step **204** which determines if E-mail delivery notification has been requested by the user. If an E-mail request has been

made by the user, then the shipping server initiates the E-mail services **100** (see Figure 4) and sends an E-mail notification to the parties specified by the user. One such party may of course be the user. This operation is shown by Step **206**. The tracking results are also stored in the database **76** by marking the package record as "DELIVERED". This information can then be used for report purposes. This is shown by Step **208**.

It is therefore seen that the automatic generation of tracking requests for undelivered packages allows the shipping server to maintain information in its database concerning the whereabouts of undelivered packages, as well as being the triggering mechanism when delivery of a package is determined, for sending E-mail notification to the sender or other designated parties. This greatly facilities the usefulness of the shipping server methodology since it augments the usefulness of the information which otherwise would have to be manually requested by the user concerning the status of a undelivered package.

In this regard, reference should again be made to Figure 7, wherein it shows that if a user tracking request (event **191**) has been made by the user (InstaTrac component **89**), then that information is presented to step **193** for prioritizing the tracking requests prior to submission of those requests to the pacing step for the tracking request at step **194**. This prioritization is actually performed by putting the user generated tracking request at the head of the input tracking queue of the designated carrier. In this way, the tracking request which otherwise would have been sent to the specified carrier, can allow for a prioritized request to be made by the user which otherwise might have to wait for a substantial period of time before presentation to the carrier. These functions are performed by the tracking coordinator **121**.